3. Botan

Program Name: Botan.java Input File: botan.dat

Poor Botan is very confused. He just learned about prime numbers, numbers with only two positive integer divisors. The first few prime numbers are 2, 3, 5, 7, 11, and 13. It's an active area of research to discover large prime numbers, as they have applications in cryptography. Botan has a solution to this problem: just change the definition of what a prime number is and call it "Botan-prime"!

To convince people to go along with the "Botan-prime", a number less than 10 is a Botan-prime if and only if it is actually prime. For numbers greater than or equal to 10, a number is "Botan-prime" if and only if all of its substrings are "Botan-prime".

A substring of a number is a continuous block of digits from that number. For example, 3892 has 9 substrings: 3, 38, 389, 3892, 8, 89, 892, 92, and 2. Those are the only substrings of 3892. A proper substring is a substring that is not the original. Thus 3892 has 8 proper substrings: 3, 38, 389, 8, 89, 892, 92, and 2.

32 is not a substring of 3892 because the digits are not consecutive. 98 is not a substring of 3892 because those digits do not appear in that order in the number. This number is not Botan-prime, as 8 is not Botan-prime.

Given a number, can you tell if it is Botan-prime or not?

Input: The first line is positive integer T (at most 20), the number of test cases to follow. Each test case contains a single positive integer N (at most 1,000,000).

Output: For each test case, print the test case number and whether the input is Botan-prime or not. Follow the formatting in the samples.

Sample input:

```
5
3
1
15
35
3892
```

Sample output:

```
Case #1: BOTAN-PRIME
Case #2: NOT BOTAN-PRIME
Case #3: NOT BOTAN-PRIME
Case #4: BOTAN-PRIME
Case #5: NOT BOTAN-PRIME
```